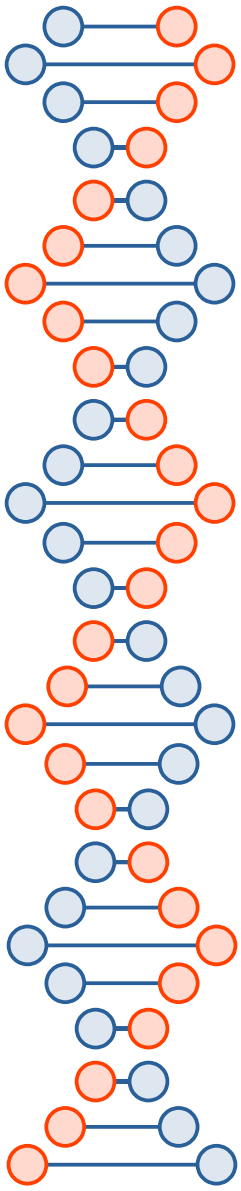
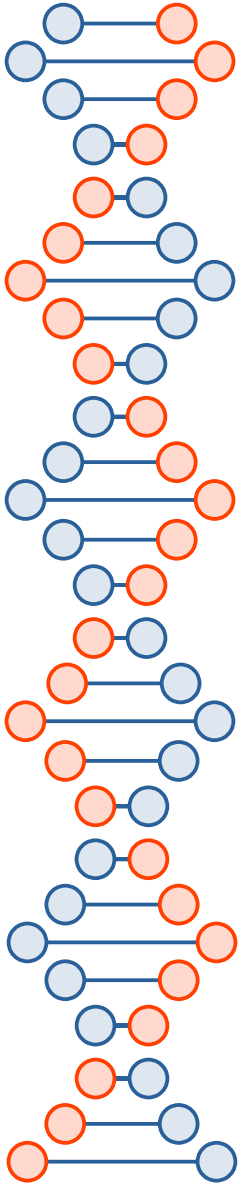


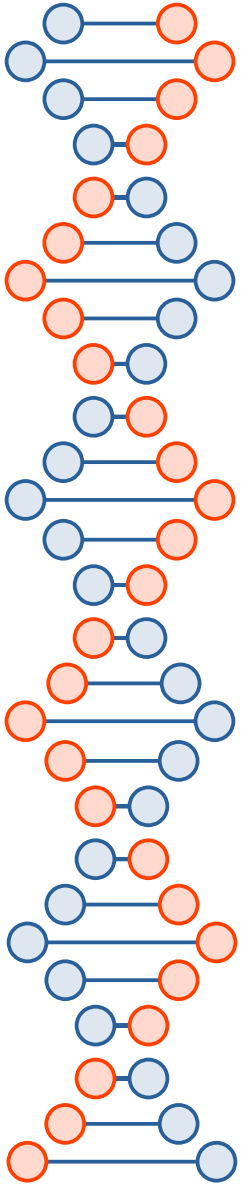
Undergraduate projects and Supervised ML

Santa Basnet,
Sr. AI Programmer
Wiseyak INC, Integrated ICT



Outline

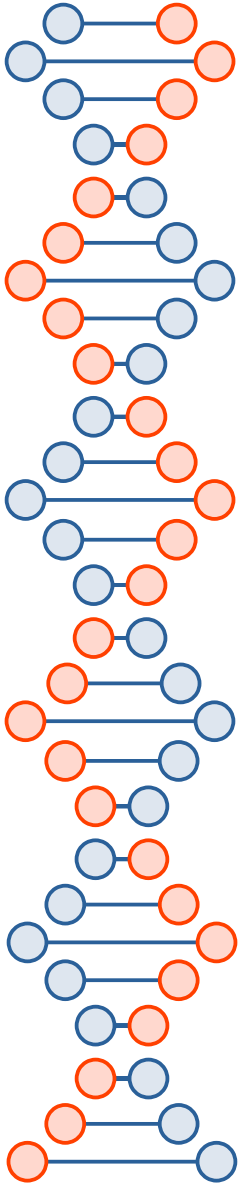
- Introduction
- ML and Supervised ML
- Some working areas
- Implementation



Introduction

- Aims to build smart systems.
- Trains data with predefined algorithms.
- Make predictions or classifies for the new inputs, especially unseen inputs.

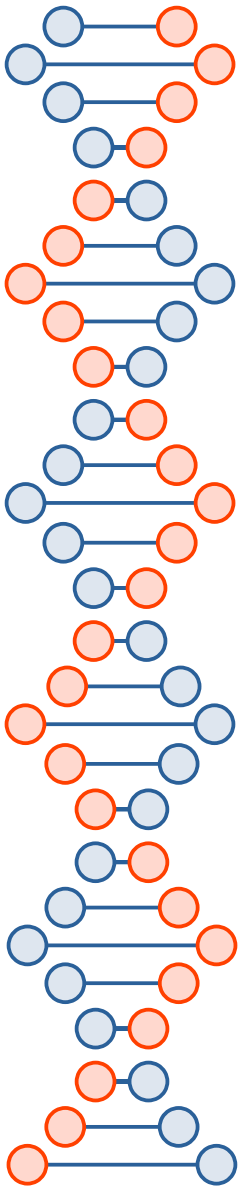
- **Machine Learning**



Introduction

- Aims to build AI models with labeled data.
- Trains data with by minimizing the prediction errors.
- Examples: Linear regression, decision trees, Naive Bayes, SVM, NN.

- **Supervised Machine Learning**



Supervised ML

- Load data:

```
x, y ← build_classification(sample = 1000, features = 10,
classes = 2)
```

- Prepare training and test data.

```
x_train, x_test, y_train, y_test ← split_train_test(x,y,
test_size=0.10)
```

- Train:

```
model ← MLAlgorithm().fit(x_train, x_test)
```

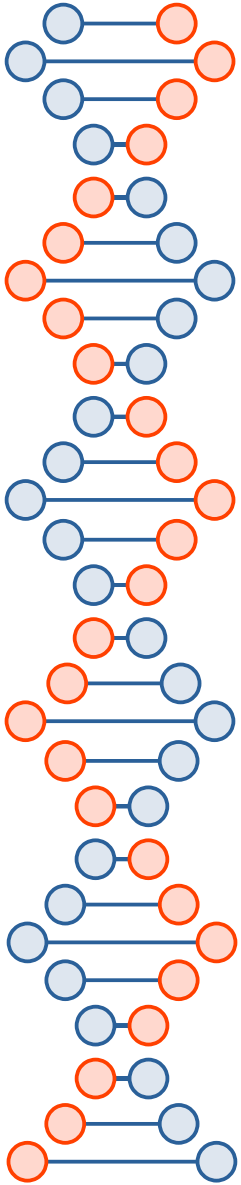
- Use:

```
y_pred <- model.predict(x_test) or model.classfy(x_test)
```

- Score:

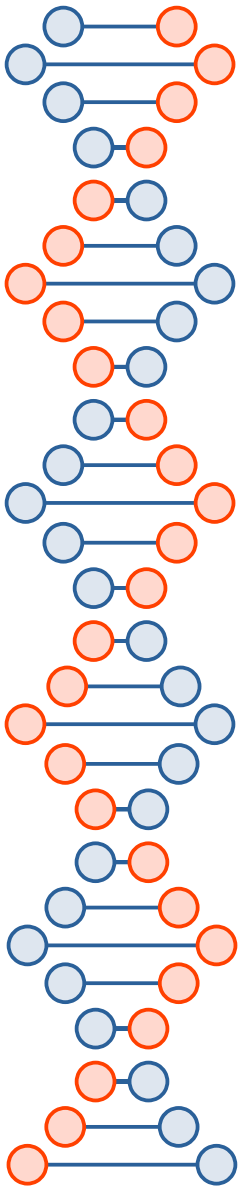
```
score ← accuracy(y_test, y_pred)
```

- **Steps**



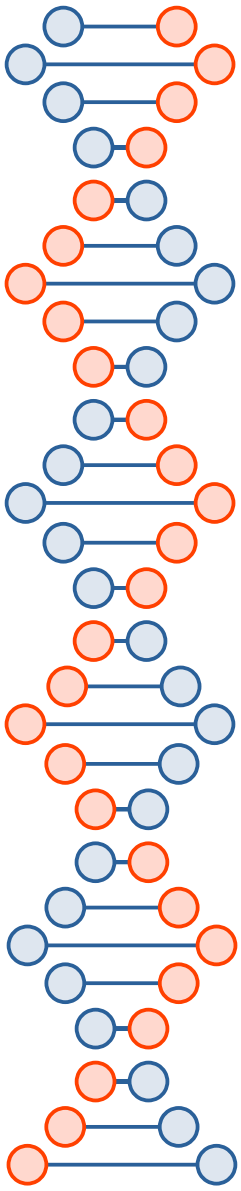
Supervised ML applications

- Image and Speech recognition.
- Medical diagnosis.
- Financial, Weather forecasting.
- Natural Language Processing.
- Network Security.
- ...



Undergraduate projects

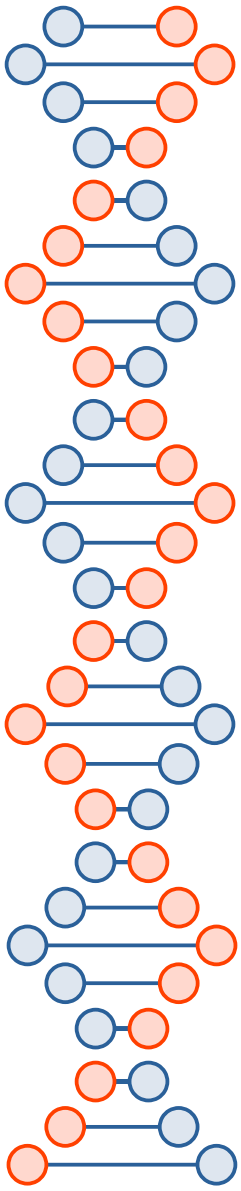
- .
- **Context of Spell**



Undergraduate projects

- .

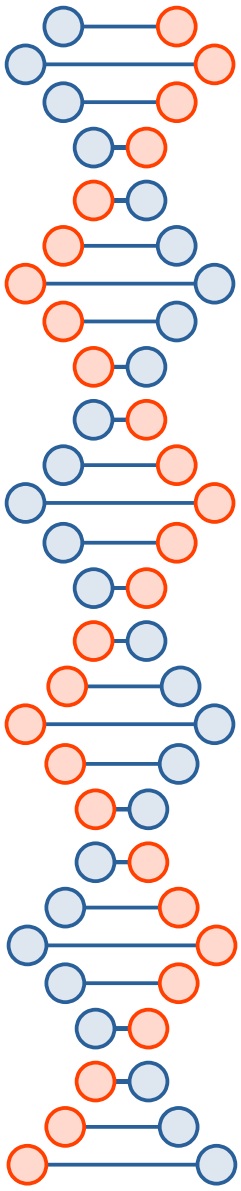
- **Scheduling and Route Planning**



Undergraduate projects

- .

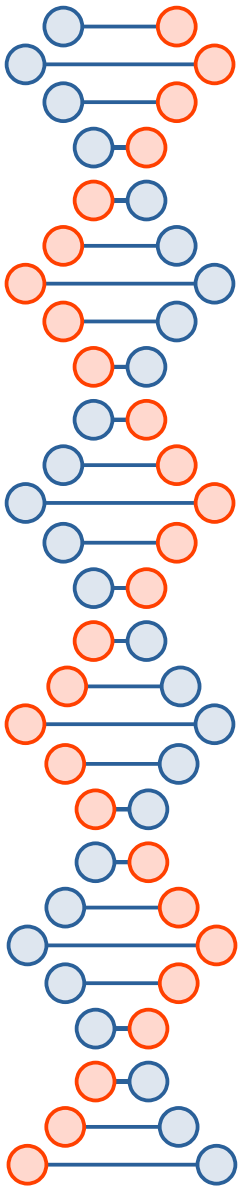
- **Graph based data modeling**



Undergraduate projects

- .

- **Query Generation from ontology graphs**



Undergraduate projects

- .

- **Smart contract based software development**

Hands on: Naive Bayes classifier

- Posterior Probability:

$$p(C_k|\vec{x}) = \frac{p(\vec{x}|C_k)p(C_k)}{p(\vec{x})} = \frac{p(x_1, ..., x_n|C_k)p(C_k)}{p(x_1, ..., x_n)}$$

where $\vec{x} = \{x_1, ..., x_n\}$ *'k = 1, …, K.*

- Using chain rule,

$$p(x_1, ..., x_n|C_k) = p(x_1|x_2, ...x_n, C_k)p(x_2|x_3, ..., x_n, C_k)...p(x_{n-1}|x_n, C_k)p(x_n|C_k)$$

- Naive Assumption:

$$p(x_i|x_{i+1}, ..., x_n|C_k) = p(x_i|C_k) \implies p(x_1, ..., x_n|C_k) = \prod_{i=1}^{n} p(x_i|C_k)$$

- **Underlying Math**

Fig: https://brilliant.org/wiki/naive-bayes-classifier/

Hands on: Naive Bayes classifier

- Finally:

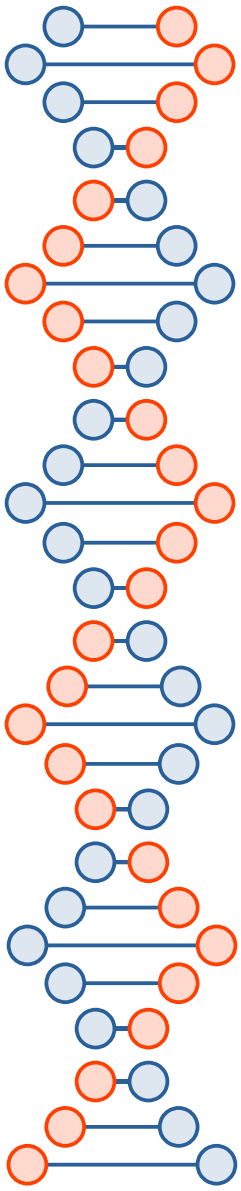
$$
\begin{aligned}
p(C_k|x_1,...,x_n) &\propto p(C_k,x_1,...,x_n) \\
&\propto p(C_k)p(x_1,...,x_n|C_k) \\
&\propto p(C_k)\ p(x_1|C_k)\ p(x_2|C_k)...p(x_n|C_k) \\
&\propto p(C_k)\prod_{i=1}^{n} p(x_i|C_k)\,.
\end{aligned}
$$

- Classification:

$$
\hat{C} = \underset{k\in\{1,...,K\}}{\operatorname{argmax}}\ p(C_k)\prod_{i=1}^{n} p(x_i|C_k)
$$

where $\hat{C}$ is the estimated class for $\vec{x}$ given its features x1, …, xn.

- **Underlying Math**



Hands on: Naive Bayes classifier

- Input:

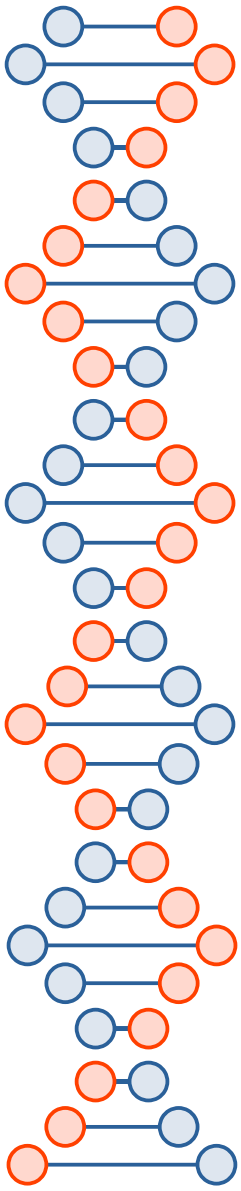
  a. Set of about 1000 words encoded in TTF fonts(Preeti, Kantipur) for Nepali language like खुवाउने, रमाए and so on.

- Output:

  a. For words : खुवाउने, रमाए >> the language is Nepali.

  b. For words : enjoy, eat >> the language is English.

- **Project Problem**



Hands on: Naive Bayes classifier

- Summary Outcome:

  Total Models : 2

  Nepali Grams Size: Probability of Language = -0.09570586672722367, Total Gram Entries : 921478

  English Grams Size: Probability of Language = -2.393946991039648, Total Gram Entries : 140760

  Classifier Results:

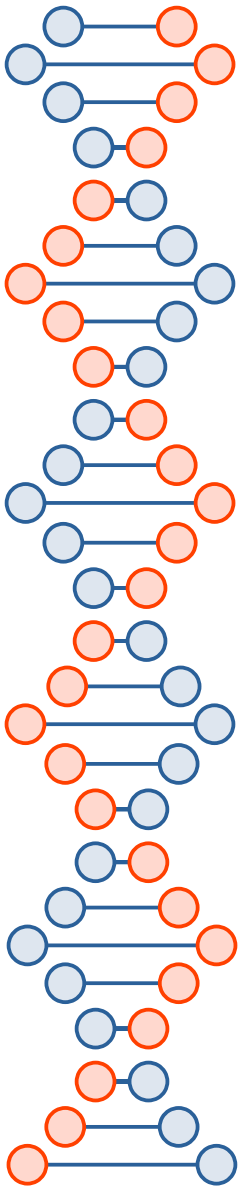
  Word = {text='software'} >> Language{code='en', name='English'}

  Word = {text=';^6jo]/'} >> Language{code='np', name='Nepali'}

  Word = {text='nepali'} >> Language{code='en', name='English'}

  Word = {text='g]kfnl'} >> Language{code='np', name='Nepali'}

  Word = {text='nationality'} >> Language{code='en', name='English'}



Undergraduate projects and Supervised ML

Questions ?